Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Estatística

Aplicação de métodos de classificação em reconhecimento facial

Caio César de Sousa Balena

Brasília

2019

Caio César de Sousa Balena

Aplicação de métodos de classificação em reconhecimento facial

Projeto apresentado para obtenção do título
de Bacharel em Estatística ao Departamento
de Estatística da Universidade de Brasília

Orientador: Prof. Dr. André Luiz Fernandes Cançado

Brasília

2019

Agradecimentos

A Deus, pela proteção e força para superar as dificuldades.

A minha esposa, Amanda, por todo apoio e compreensão durante a realização desse curso.

A meus pais, Carlos e Sônia, e meu irmão Huan, pela motivação e orações.

Ao professor Dr. André Luiz Fernandes Cançado, pela atenção e disposição em me orientar nesse trabalho.

A meus amigos e colegas, Adolfo, Allan Vieira, Eduardo Hellas, Frederico, Pedro Brom, Rômulo Coutinho, que fizeram parte dessa caminhada.

A Universidade de Brasília e todos os docentes que fizeram parte dessa jornada.

Por fim , a todos que contribuíram de alguma forma para conclusão dessa etapa.

Resumo

O reconhecimento facial é uma tecnologia aplicada com diversos propósitos, desde controle de acesso a dispositivos a identificação de potenciais ameaças a segurança da sociedade. Nesse trabalho é proposto a redução de uma base de dados composta por imagens de faces através de análise de componentes principais e em seguida é desenvolvido um algoritmo capaz de classificar essas imagens somente pela distância euclidiana entre essas imagens projetadas em um subespaço vetorial com dimensão bastante inferior ao original. Para o caso em que o algoritmo classifica imagens de grupos utilizados para treinamento obteve-se resultados bastante interessantes. Para o caso de classificação de imagens não pertencentes aos grupos treinados houve uma considerável queda nas taxas de acerto.

Palavras-chave: análise de componentes principais, distância euclidiana, classificação, Monte Carlo Cross Validation.

Sumário

1 Introdução

A utilização de computadores para realizar a identificação de indivíduos é uma aplicação bastante empregada atualmente. Essa ferramenta se encontra em uso em aeroportos, transportes coletivos, smartphones entre outros, sendo que sua principal aplicação está relacionada à área de segurança.

Essa identificação pode ser realizada através da mensuração de diversas características de uma pessoa, entre elas estão: a face, as impressões digitais, a íris, a voz, etc. E cada tipo de característica possui vantagens e desvantagens.

Nesse trabalho será abordado o reconhecimento de pessoas através da face. Uma das vantagens desse tipo de identificação está relacionada ao fato de não ser invasiva. Outros trabalhos já foram desenvolvidos na área de reconhecimento facial (KIM, 1996) e serviram de base para o desenvolvimento desse.

O problema de identificar indivíduos é abordado pela estatística como um problema de classificação. Nesse caso, cada grupo é uma pessoa e o algoritmo classificador precisar alocar novas imagens nos grupos treinados, ou rejeitá-las, caso não pertençam a nenhum dos grupos.

Uma dificuldade presente em problemas desse tipo reside no fato de que a base de dados, as imagens, formam um conjunto de dados em espaços com dimensões muito grandes. Para resolução desse problema é possível utilizar análise de componentes principais. Dentre os objetivos da aplicação desse tipo de análise encontra-se a redução de dimensionalidade (JOHNSON, 2012).

1.1 Objetivos

1.1.1 Objetivo geral

O objetivo desse trabalho está em desenvolver um algoritmo capaz de realizar a classificação de imagens através da avaliação da distância euclidiana entre os escores das imagens. Essa tarefa será realizada em duas etapas: a primeira consiste em classificar apenas imagens de grupos utilizados para treinamento. A próxima etapa apresentará uma adaptação do primeiro algoritmo que permitirá a rejeição de imagens que não pertençam a nenhuma classe.

1.1.2 Objetivos específicos

- Implementar no software R um método de classficação de imagens baseado em PCA;
- Aplicar o método em uma base de dados real; e
- Avaliar a acurácia do método em cada caso de teste.

2 Base de dados

A base de dados empregada nesse trabalho foi obtida no sítio da *AT&T Laboratories Cambridge*. Nessa base há dez imagens para cada um dos quarenta indivíduos, totalizando 400 fotos. Cada pessoa corresponde a um grupo, portanto, há 40 grupos. As imagens possuem 92×112 *pixels* e estão em escala de cinza.

Abaixo é possível visualizar algumas das imagens que compõem a base de dados.

Antes de aplicar a técnica proposta é necessário converter as imagens para um formato que as torne tratáveis. Cada imagem é uma matriz de *pixels*. Portanto, as imagens foram convertidas em vetores e agrupadas em uma base de dados única, contendo 400 linhas e 10304 colunas, ao final desse processo, para cada observação, foi acrescentado uma nova variável, um número que representa o grupo ao qual a imagem pertence..

Para realizar esse tratamento inicial das imagens foi utilizada a função *"load.image()"* do pacote *"imager"*, do *software* R. O processo de conversão atribui um número entre 0 e 1 para cada *pixel*. Valores próximos a zero indicam cores próximas ao preto e valores próximos a um indicam cores próximas ao branco.

3 Análise de Componentes Principais

A análise de componentes principais (Principal Components Analysis – PCA) foi introduzida por Karl Pearson em 1901 e fundamentada por Hotelling em 1933. O objetivo dessa técnica consiste em explicar a estrutura de variância e covariâncias de um vetor aleatório de dimensão p através de combinações lineares das variáveis originais. Em geral, ao aplicar essa técnica em um conjunto de dados espera-se obter a redução da dimensão desses dados e interpretação das combinações lineares construídas (MINGOTI, 2005).

Embora as p componentes sejam necessárias para compor toda a variabilidade de um conjunto de dados, em muitas situações grande parte dessa variabilidade pode ser explicada por uma pequena quantidade k ($k < p$) das componentes principais. Ao obter combinações lineares $Z_1, Z_2, \ldots, Z_p$ a partir das variáveis originais $X_1, X_2, \ldots, X_p$, estaremos obtendo um novo sistema de variáveis não correlacionadas que estará ordenado por importância, ou seja, $Var(Z_1) \geq Var(Z_2) \geq \cdots \geq Var(Z_p)$. Assim, espera-se que a variabilidade explicada pela maioria das combinações seja tão pequena que possam ser desprezadas, obtendo dessa forma uma redução da dimensão dos dados (MANLY, 2016).

O desenvolvimento da PCA não requer que os dados possuam distribuição normal multivariada, porém, inferências podem ser feitas quando as componentes são extraídas de amostras com essa distribuição (JOHNSON, 2012).

3.1 Execução da análise de componentes principais

Seja um vetor aleatório **X'** = $[X_1,\ X_2,\ \ldots,\ X_p]$ com matriz de covariâncias $\sum_{p\times p}$. Sejam $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p > 0$ os autovalores da matriz $\sum_{p\times p}$, com os respectivos autovetores normalizados $a_1, a_2, \ldots, a_p$, isto é, os autovetores a_i satisfazem as seguintes condições:

(i) $a_i' a_j = 0$ para todo $i \neq j$;

(ii) $a_i' a_i = 1$ para todo $i = 1, 2, \ldots, p$;

(iii) $\sum_{p\times p} a_i = \lambda_i a_i$, para todo $i = 1, 2, \ldots, p$.

sendo o autovetor a_i denotado por $a_i = (a_{i1}\ a_{i2}\ \ldots\ a_{ip})'$. Considere o vetor aleatório $Z = O'X$, no qual $O_{p\times p}$ é a matriz ortogonal composta pelos autovetores normalizados da matriz $\sum_{p\times p}$, isto é,

$$O_{p\times p} = \begin{bmatrix} a_{11} & a_{21} & \ldots & a_{p1} \\ a_{12} & a_{22} & \ldots & a_{p2} \\ \cdot & \cdot & \ldots & \cdot \\ \cdot & \cdot & \ldots & \cdot \\ \cdot & \cdot & \ldots & \cdot \\ a_{1p} & a_{2p} & \ldots & a_{pp} \end{bmatrix} = [a_1\ a_2\ \ldots\ a_p].$$

O vetor Z é composto de p combinações lineares das variáveis aleatórias do vetor **X'**, possui vetor de médias $O'\mu$, em que μ é o vetor de médias de **X'**, e matriz de covariâncias $\Lambda_{p\times p}$, que é uma matriz diagonal, isto é,

$$\Lambda_{p\times p} = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \lambda_p \end{bmatrix}.$$

Logo, as variáveis aleatórias que constituem o vetor Z são não correlacionadas entre si. Os vetores aleatórios X e Z possuem a mesma variância, porém, o vetor Z tem a vantagem de ser composto por variáveis não correlacionadas, possibilitando assim, uma redução do espaço de variáveis.

A seguir são apresentadas algumas definições:

Definição 1: A j-ésima componente principal da matriz $\sum_{p\times p}$ é representada por:

$$Z_j = a_j'\mathbf{X} = a_{j1}X_1 + a_{j2}X_2 + \cdots + a_{jp}X_p \quad j = 1, 2, \ldots, p$$

A esperança e variância da componente Z_j são, respectivamente, iguais a :

$$E[Z_j] = a_j'\mu = a_{j1}\mu_1 + a_{j2}\mu_2 + \cdots + a_{jp}\mu_p$$

e

$$Var[Z_j] = a_j' \sum a_j$$

sendo $Cov(Z_i, Z_j) = 0, i \neq j$. Cada autovalor λ_i representa a variância de uma componente principal, e como os autovalores estão ordenados por importância, a primeira componente é a de maior variabilidade e a p-ésima é a de menor.

Definição 2: A proporção da variância de $\mathbf{X}$ explicada pela j-ésima componente principal é definida como:

$$\frac{Var(Z_j)}{Variância\ total\ de\ X} = \frac{\lambda_j}{\sum_{i=1}^{p} \lambda_i}$$

Visando impedir que uma variável ou grupo de variáveis venha exercer grande influência nas componentes principais, é comum padronizar as variáveis para que tenham média zero e variância unitária (MANLY, 2016). Dessa forma, a PCA será aplicada na matriz de correlação de $\mathbf{X}$.

A seguir estão apresentados os passos para realização da análise de componentes principais (MANLY, 2016):

1. Padronizar as variáveis $X_1, X_2, \ldots, X_p$;
2. Calcular a matriz de covariâncias, ou de correlações;
3. Obter os autovalores $\lambda_1, \lambda_2, \ldots, \lambda_p$ e os respectivos autovetores associados $a_1, a_2, \ldots, a_p$. Os coeficientes da i-ésima componente principal são os elementos do autovetor a_i, enquanto λ_i é a sua variância.
4. Descartar as componentes responsáveis por uma quantidade ínfima de variabilidade.

3.2 Aplicação da PCA na base de dados

Uma vez definida a quantidade de componentes principais a serem utilizadas, esses autovetores selecionados formarão uma base ortonormal na qual as imagens poderão ser representadas em um subespaço vetorial com uma dimensão muito inferior à original. Supondo que k componentes foram selecionadas, uma imagem será representada da forma

$$\Omega = [\omega_1 \omega_2 \ldots \omega_k]',$$

onde:

$$\begin{aligned}
\omega_1 &= a_1'\mathbf{X} = a_{11}X_1 + a_{12}X_2 + \cdots + a_{1p}X_p \\
\omega_2 &= a_2'\mathbf{X} = a_{21}X_1 + a_{22}X_2 + \cdots + a_{2p}X_p \\
&\vdots \\
\omega_k &= a_k'\mathbf{X} = a_{k1}X_1 + a_{k2}X_2 + \cdots + a_{kp}X_p
\end{aligned}$$

Esses escores serão calculados para a base de treinamento e para a de teste e a classificação ocorrerá em razão da distância euclidiana entre cada observação de teste e de treinamento.

Na próxima seção serão descritos os procedimentos para realização das classificações.

4 Classificação por Distância Euclidiana

Antes da aplicação da PCA é necessário realizar a partição da base de imagens em dados de treinamento, sobre os quais a técnica será aplicada, e dados de teste, para estimação da acurácia do classificador.

Suponha que após a aplicação da PCA foram selecionadas k componentes para representar as imagens. Então, serão calculados os escores para as imagens de treinamento, Ω_t e de teste, Ω_v. A distância euclidiana d entre uma imagem de teste e de treinamento, é dada por:

$$d = \|(\Omega_v - \Omega_t)\| = \sqrt{\sum_{i=1}^{k} (\Omega_{v_i} - \Omega_{t_i})^2}$$

As subseções a seguir visam descrever os algoritmos de classificação desenvolvidos.

4.1 Classificação de imagens de grupos treinados

Nesse primeiro caso em estudo, o algoritmo proposto realizará apenas a classificação de imagens que pertençam a algum dos grupos utilizados durante o treinamento.

O classificador recebe como parâmetros de entrada a base de dados das imagens, a quantidade de componentes principais a serem extraídas e um vetor de inteiros que será utilizado para separar a base em treinamento e teste. Os procedimentos realizados pelo algoritmo estão descritos a seguir:

1. O algoritmo separa a base de dados em treinamento e teste utilizando o vetor de inteiros como referência para partição;

2. É aplicada a análise de componentes principais na base de treinamento. São extraídos os k autovetores definidos no início do algoritmo;

3. São obtidos os escores para as imagens de treinamento e de teste;

4. Realiza-se o cálculo das distâncias entre os escores das imagens de teste e de treinamento. Esses resultados são armazenados em uma matriz na qual as linhas representam as imagens de teste e as colunas representam as imagens de treinamento. É realizada a avaliação dos menores valores de cada linha dessa matriz, as posições desses mínimos são registradas em um vetor;

5. É realizada a classificação das imagens de teste. Esse resultado é salvo em um vetor. O teste é realizado a partir da comparação do valor da variável resposta da imagem de teste com o valor da resposta da imagem de treinamento para a qual foi obtida a menor distância. Para cada classificação correta é atribuído valor um ao vetor de acertos, caso contrário é atribuído valor zero.

6. O último passo consiste em calcular a acurácia a partir da proporção de classificações corretas realizadas pelo algoritmo.

4.2 Classificação de imagens de grupos não treinados

Nesse caso o algoritmo deve classificar também imagens que não pertençam a nenhum dos grupos utilizados no treinamento. A solução proposta para esse problema consiste em definir um limite para cada grupo treinado, de acordo com a ideia a seguir:

Suponha que sejam utilizadas imagens de m classes para treinamento, e que em cada classe há n imagens para treino. A PCA será aplicada nessa base composta por $m \times n$ observações e os escores serão calculados. A seguir serão definidas as distâncias limite para grupo treinado. Para cada grupo será realizado o seguinte procedimento:

1. Calcula-se as $\binom{n}{2}$ distâncias possíveis entre as imagens de treinamento do grupo, de forma a obter uma distribuição empírica de distâncias; e

2. Define-se a distância limite para o grupo como o q-ésimo percentil dessa distribuição obtida no passo anterior.

Logo, haverá m limites de classificação, um para cada classe utilizada no treinamento. As imagens que não forem classificadas em nenhum grupo serão designadas como pertencentes a uma classe m+1, de imagens desconhecidas.

Essa versão do classificador recebe como parâmetros de entrada a base de dados de imagens, a quantidade de componentes principais a serem extraídas, um vetor de inteiros que será utilizado para separar a base em treinamento e teste e o percentil a ser utlizado para definição dos limites de classificação para cada grupo. Os procedimentos realizados pelo algoritmo estão descritos a seguir:

1. O algoritmo separa a base de dados em treinamento e teste utilizando o vetor de inteiros como referência para partição;

2. É aplicada a análise de componentes principais na base de treinamento. São extraídas os k autovetores definidos no início do algoritmo;

3. São obtidos os escores para as imagens de treinamento e de teste;

4. São definidos os limites de classificação para cada grupo treinado;

5. Realiza-se o cálculo das distâncias entre os escores das imagens de teste e de treinamento. Esses resultados são armazenados em uma matriz na qual as linhas representam as imagens de teste e as colunas representam as imagens de treinamento;

6. É realizada a classificação das imagens de teste. Para cada imagem de teste o algoritmo ordena as distâncias em ordem crescente e realiza a comparação até alocar essa observação em um grupo ou no grupo de imagens desconhecidas. Da mesma maneira como ocorre no classificador anterior, o resultado é salvo em um vetor em que valores 1 representam acertos e 0 representam erros;

7. O último passo consiste em calcular a acurácia para o grupo de imagens pertencentes a grupos usados para treinamento e uma outra acurácia para as imagens que não pertençam a nenhum grupo utilizado para treinamento.

5 Precisão das estimativas

Para avaliar o desempenho de um método de aprendizado estatístico em um dado conjunto de dados é necessário utilizar ferramentas para quantificar a precisão das estimativas obtidas (JAMES, 2013).

Para isso, foi empregado nesse trabalho o método definido como *Monte Carlo Cross Validation - MCCV* que consiste em particionar a base de dados aleatoriamente em grupo de treinamento e grupo de teste n vezes. Para cada particionamento um modelo é ajustado e tem sua acurácia estimada com os dados de teste (XU, 2001).

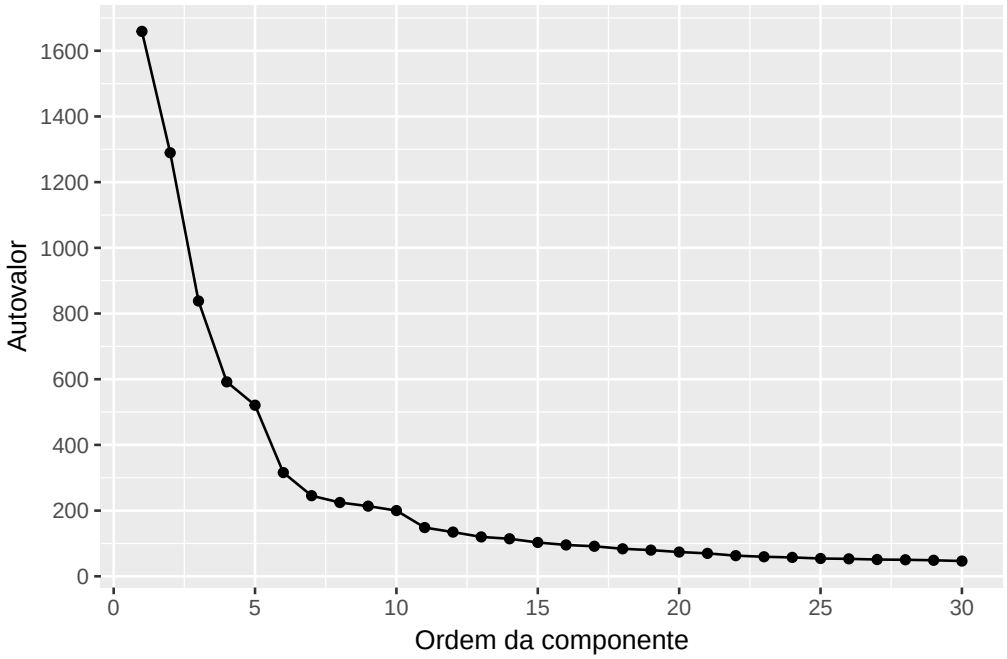
Considerando que para cada caso em estudo sejam geradas n repetições do classificador, então para cada caso a acurácia será a média das estimativas obtidas e a precisão será dada em função do erro padrão das estimativas.

6 Resultados

Antes de iniciar as execuções dos algoritmos é necessário defenir a quantidade de componentes principais a serem utilizadas no cálculo dos escores de cada imagem.

Uma das formas de defenir a quantidade de componentes principais está no uso do *scree plot*, que mostra os valores numéricos dos autovalores $\hat{\lambda}_i$ de acordo com a ordem da respectiva componente. Basta observar no gráfico o ponto em que os valores de $\hat{\lambda}_i$ tendem a se estabilizar (MINGOTI, 2005). Abaixo é possível observar o *scree plot* dos primeiros 30 autovalores da base de imagens.

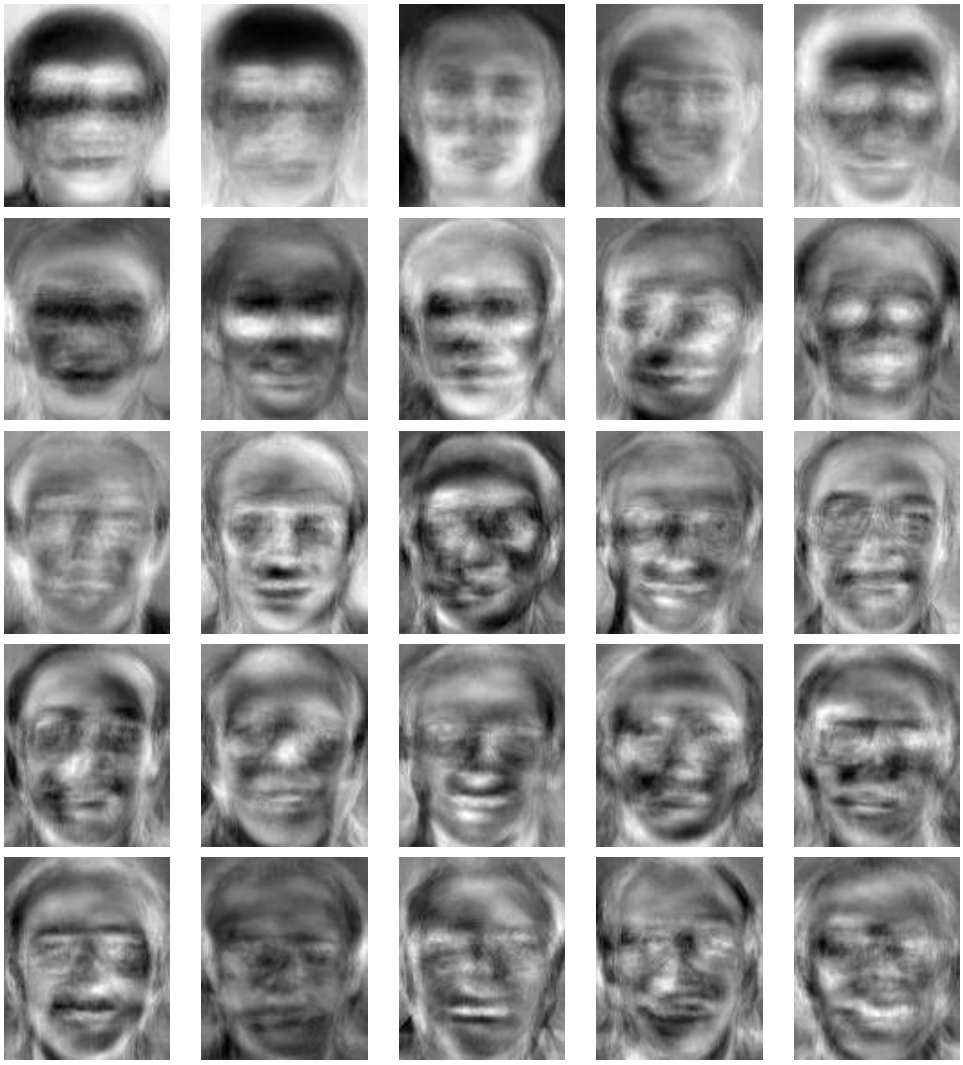
Gráfico 1: Scree plot



Observando o gráfico acima percebe-se que a partir da 11ª componente os valores de $\hat{\lambda}_i$ tendem a se estabilizar, então, a escolha de uma quantidade de componentes a partir de 11 pode apresentar bons resultados para os testes a seguir.

Para ambos os casos de teste foram utilizadas 10, 15, 20 e 25 componentes para realização das classificações. Conforme mencionado na seção anterior, o método de verificação da precisão da acurácia será a partir do *Monte Carlo Cross Validation*. Para cada caso em estudo foram realizadas 1000 repetições e a acurácia estimada é dada em função da média das acurácias obtidas.

A seguir é possível visualizar as 25 primeiras componentes principais reconstruídas como imagens.



6.1 Classificação de imagens de grupos treinados

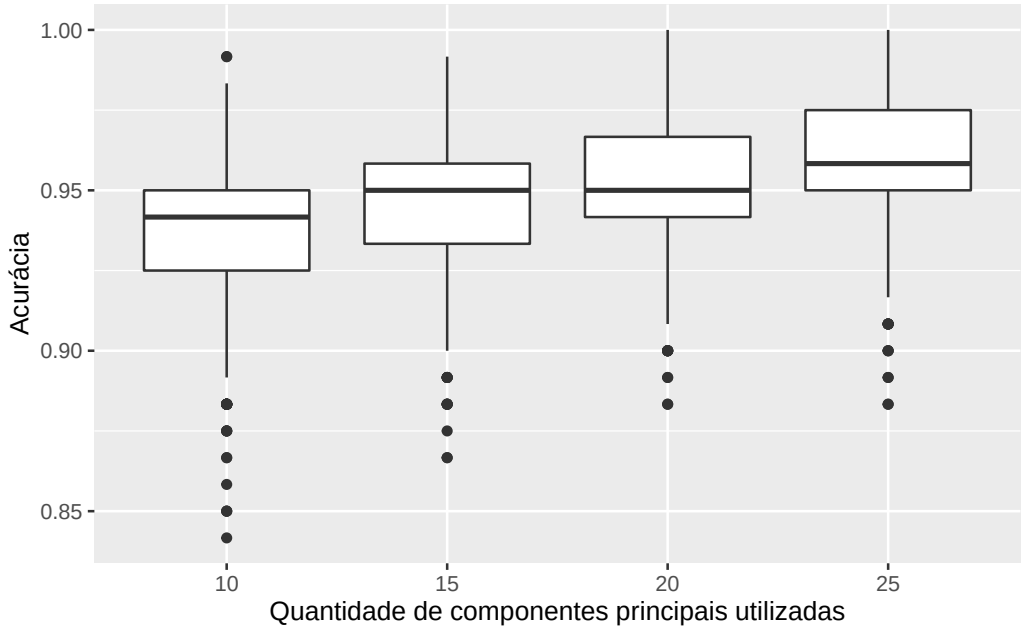
Nesse momento o classificador testou apenas imagens que pertenciam a algum dos grupos utilizados no treinamento. Foram considerados os 40 grupos disponíveis na base de dados para a execução do algoritmo. Foi utilizada 70% da base de dados para treinamento e o restante para teste.

A tabela 1 apresenta a acurácia média e o erro padrão considerando diferentes quantidades de componentes principais.

Tabela 1: Acurácia média e erro padrão para o primeiro caso de teste.

componentes	média	erro padrão
10 componentes	0.93704	0.00002
15 componentes	0.94652	0.00002
20 componentes	0.95264	0.00002
25 componentes	0.95995	0.00002

Gráfico 2: Acurácias estimadas



A acurácia para esse caso de teste apresentou resultados que convergem para 95% de acurácia, poucas repetições abaixo de 90% e em alguns casos todas as predições foram corretas. Verifica-se que o aumento na quantidade de componentes principais utilizadas implicou um aumento, contido, na acurácia média.

A seguir serão apresentados os resultados para o caso no qual o classificador classifica imagens que não pertencem a grupos utilizados para treinamento.

6.2 Classificação de imagens de grupos não treinados

Conforme mencionado na seção 4.2 o classificador define distâncias limite em cada classe a partir de um percentil da distribuição empírica das distâncias entre as imagens de cada grupo.

A avaliação dessa versão do classificador foi realizada variando os seguintes parâmetros: quantidade de componentes principais utilizadas e o percentil empregado para definição das distâncias limite de cada grupo. Para cada combinação desses parâmetros foram executadas 1000 repetições.

As tabelas a seguir apresentam as acurácias médias e seus erros padrão considerando o cenário no qual é realizada a classificação de imagens que pertencem a algum grupo utilizado no treinamento A análise dos resultados para esses testes foi dividida em duas partes. A primeira analisa o resultado da classificação das imagens pertencentes aos grupos treinados e a segunda, a classificação das imagens pertencentes aos grupos não utilizados para treinamento.

Tabela 2: Acurácia estimada para a classificação de imagens pertencentes aos grupos treinados

	10 componentes		15 componentes		20 componentes		25 componentes	
quantil	acurácia	erro padrão	acurácia	erro padrão	acurácia	erro padrão	acurácia	erro padrão
0.05	0.0840	0.0001	0.0857	0.0001	0.0885	0.0001	0.0921	0.0001
0.10	0.2172	0.0002	0.2106	0.0002	0.2080	0.0002	0.2115	0.0002
0.15	0.3830	0.0003	0.3475	0.0003	0.3558	0.0003	0.3752	0.0003
0.20	0.5479	0.0003	0.5126	0.0003	0.5289	0.0003	0.5356	0.0003
0.25	0.6746	0.0003	0.6337	0.0003	0.6423	0.0003	0.6582	0.0003
0.30	0.7445	0.0003	0.7442	0.0003	0.7596	0.0003	0.7780	0.0002
0.35	0.8204	0.0002	0.8184	0.0002	0.8399	0.0002	0.8583	0.0002
0.40	0.8570	0.0002	0.8701	0.0002	0.8831	0.0002	0.9008	0.0001
0.45	0.8708	0.0002	0.9093	0.0001	0.8644	0.0002	0.9267	0.0001
0.50	0.8994	0.0002	0.9242	0.0001	0.8948	0.0002	0.9440	0.0001
0.55	0.9178	0.0001	0.9346	0.0001	0.9171	0.0001	0.9530	0.0001
0.60	0.9257	0.0001	0.9385	0.0001	0.9207	0.0002	0.9595	0.0001
0.65	0.9259	0.0002	0.9410	0.0001	0.9259	0.0002	0.9611	0.0001
0.70	0.9291	0.0002	0.9440	0.0001	0.9291	0.0002	0.9626	0.0001
0.75	0.9316	0.0002	0.9451	0.0001	0.9316	0.0002	0.9638	0.0001
0.80	0.9315	0.0002	0.9483	0.0001	0.9315	0.0002	0.9635	0.0001
0.85	0.9370	0.0001	0.9507	0.0001	0.9370	0.0001	0.9643	0.0001
0.90	0.9389	0.0001	0.9511	0.0001	0.9389	0.0001	0.9639	0.0001
0.95	0.9391	0.0001	0.9514	0.0001	0.9391	0.0001	0.9628	0.0001

Tabela 3: Acurácia estimada para a classificação de imagens pertencentes aos grupos não utilizados para treinamento

	10 componentes		15 componentes		20 componentes		25 componentes	
quantil	acurácia	erro padrão	acurácia	erro padrão	acurácia	erro padrão	acurácia	erro padrão
0.05	0.6835	0.0005	0.7885	0.0004	0.8616	0.0003	0.8760	0.0003
0.10	0.2766	0.0004	0.2454	0.0004	0.3310	0.0005	0.3770	0.0005
0.15	0.0412	0.0002	0.0379	0.0001	0.0510	0.0001	0.0651	0.0002
0.20	0.0266	0.0000	0.0376	0.0000	0.0458	0.0000	0.0444	0.0000
0.25	0.0486	0.0000	0.0552	0.0000	0.0579	0.0000	0.0541	0.0000
0.30	0.0649	0.0000	0.0693	0.0000	0.0661	0.0000	0.0604	0.0000
0.35	0.0714	0.0000	0.0690	0.0000	0.0646	0.0000	0.0608	0.0000
0.40	0.0682	0.0000	0.0635	0.0000	0.0601	0.0000	0.0555	0.0000
0.45	0.0478	0.0000	0.0389	0.0000	0.0478	0.0000	0.0393	0.0000
0.50	0.0422	0.0000	0.0315	0.0000	0.0416	0.0000	0.0332	0.0000
0.55	0.0348	0.0000	0.0239	0.0000	0.0346	0.0000	0.0274	0.0000
0.60	0.0268	0.0000	0.0205	0.0000	0.0276	0.0000	0.0241	0.0000
0.65	0.0224	0.0000	0.0191	0.0000	0.0224	0.0000	0.0219	0.0000
0.70	0.0189	0.0000	0.0190	0.0000	0.0189	0.0000	0.0244	0.0000
0.75	0.0177	0.0000	0.0200	0.0000	0.0177	0.0000	0.0238	0.0000
0.80	0.0186	0.0000	0.0209	0.0000	0.0186	0.0000	0.0231	0.0000
0.85	0.0198	0.0000	0.0231	0.0000	0.0198	0.0000	0.0239	0.0000
0.90	0.0225	0.0000	0.0257	0.0000	0.0225	0.0000	0.0257	0.0000
0.95	0.0278	0.0000	0.0295	0.0000	0.0278	0.0000	0.0272	0.0000

A apresentação gráfica mostra a distribuição dos resultados para cada ciclo de repetições. Cada gráfico apresenta os resultados para um percentil utilizado para construção da regra de aceitação. Essa apresentação permite avaliar o comportamento do classificador a medida que se altera a quantidade de componentes principais utilizada. Os gráficos apresentam *boxplots* justapostos para verificar a acurácia ao classificar imagens que não pertencem a nenhum grupo (à esquerda) e imagens que pertencem a algum grupo (à direita).

Gráfico 3: Acurácia estimada sob o quantil 0.05

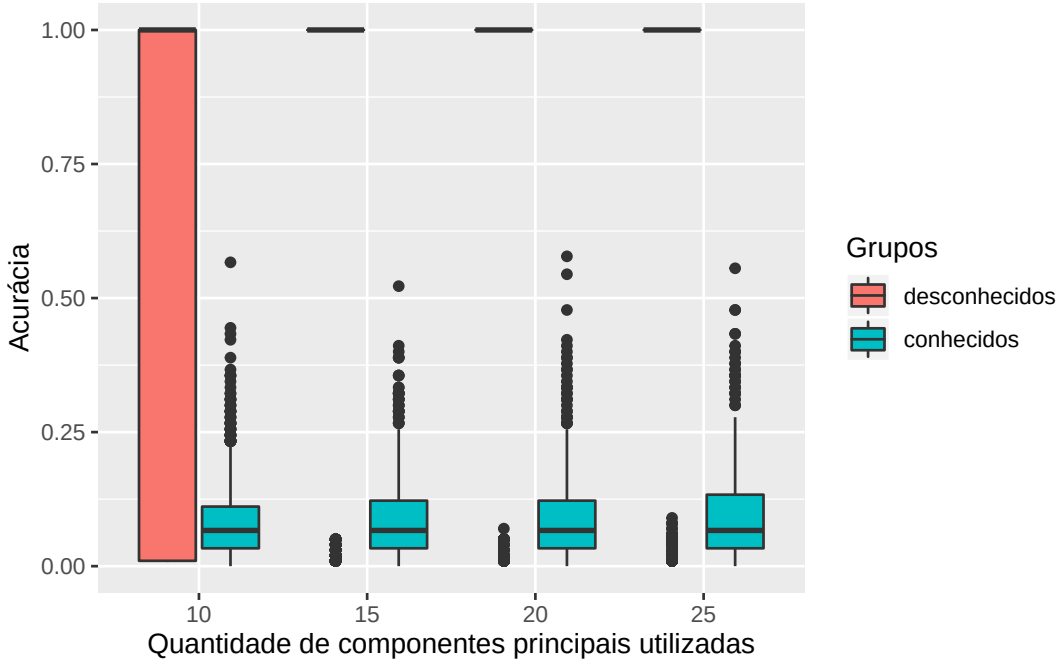


Gráfico 4: Acurácia estimada sob o quantil 0.10

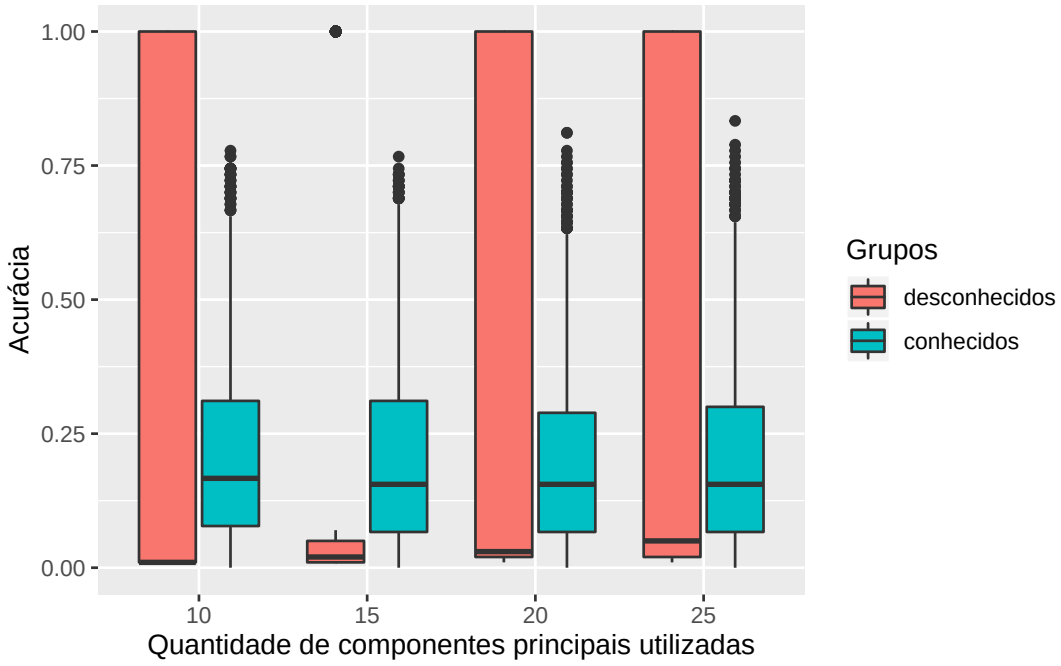


Gráfico 5: Acurácia estimada sob o quantil 0.15

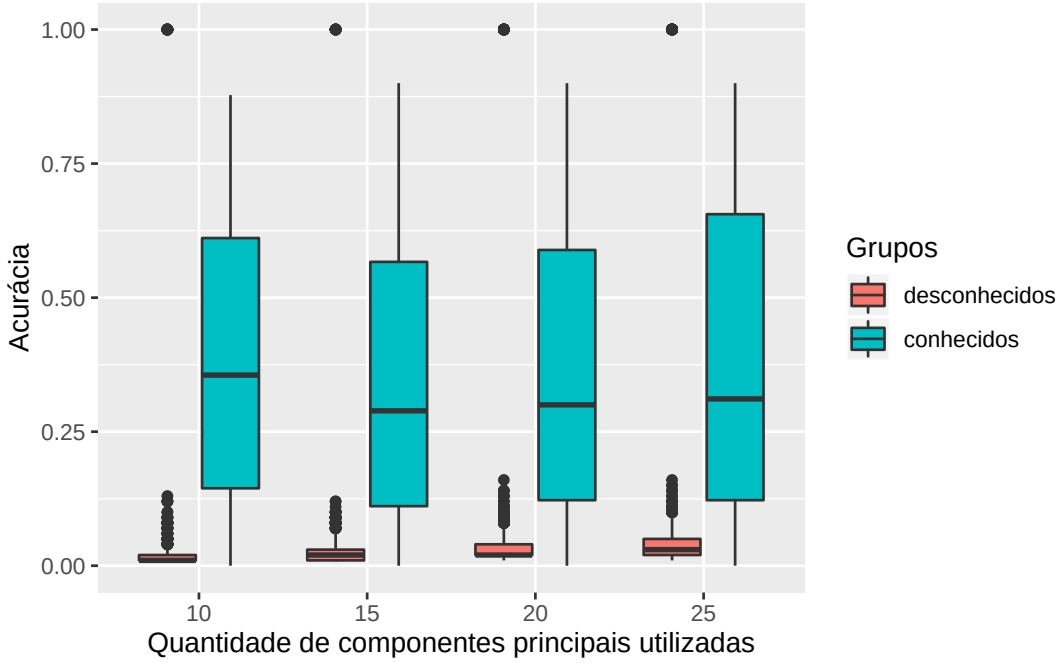


Gráfico 6: Acurácia estimada sob o quantil 0.20

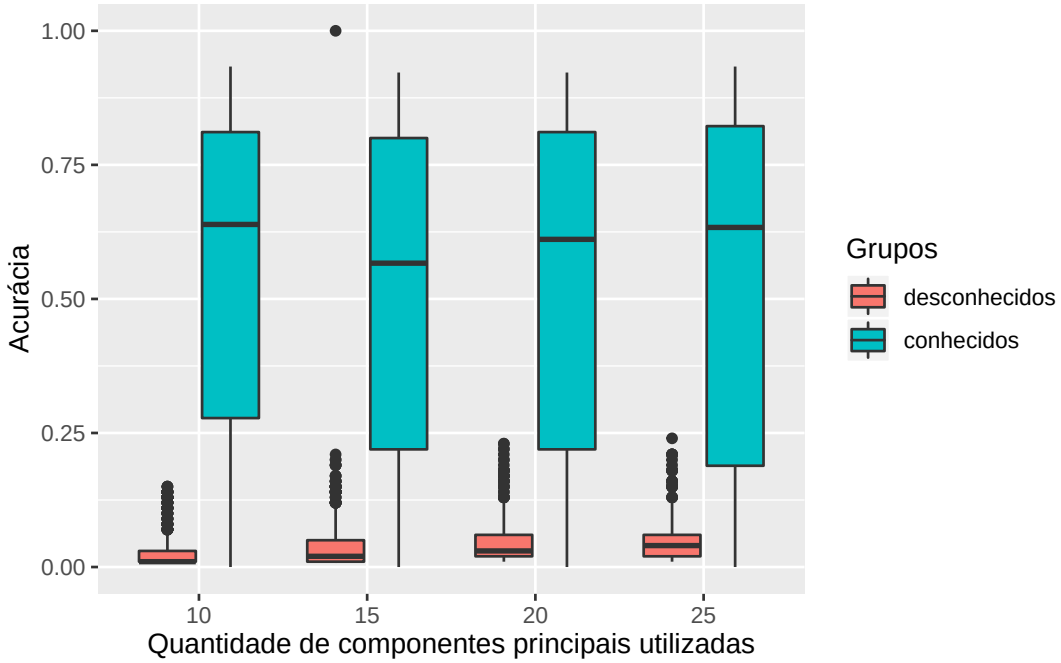


Gráfico 7: Acurácia estimada sob o quantil 0.25

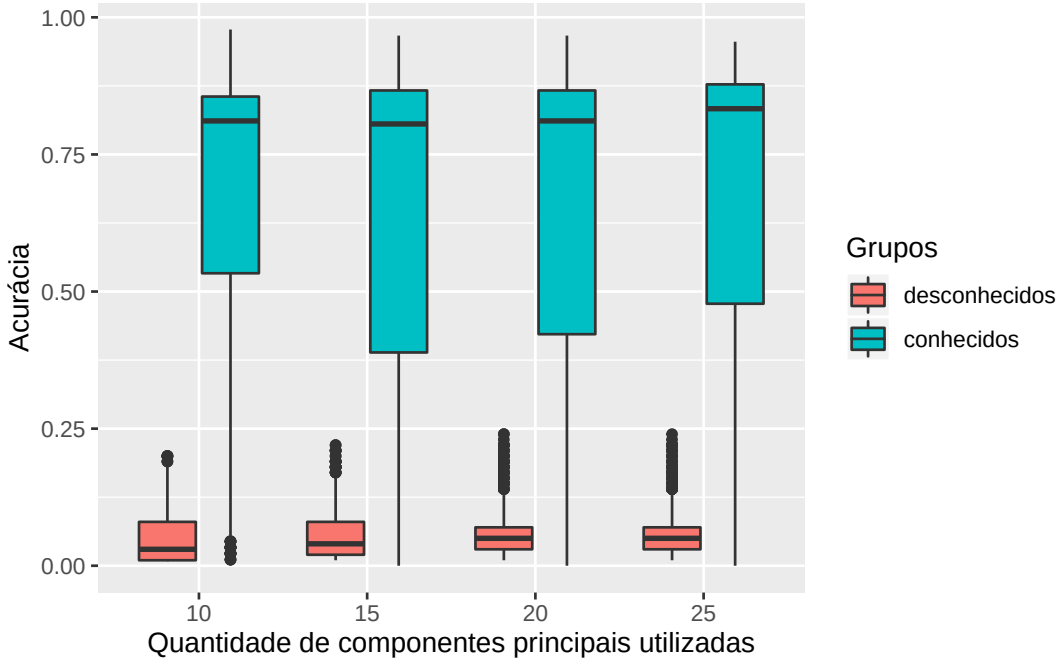


Gráfico 8: Acurácia estimada sob o quantil 0.30

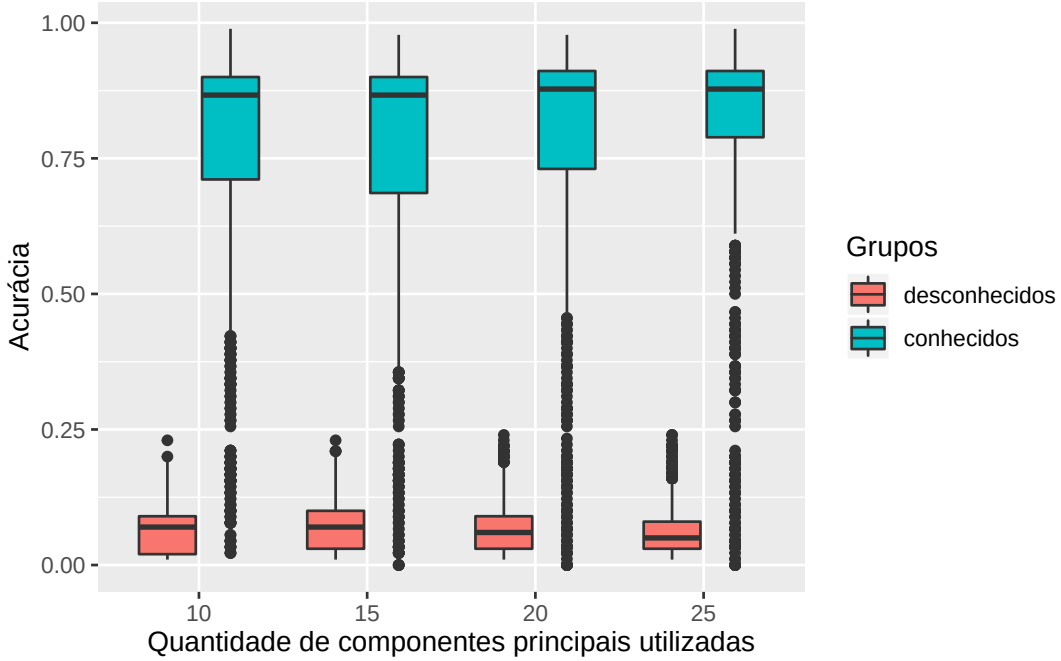


Gráfico 9: Acurácia estimada sob o quantil 0.35

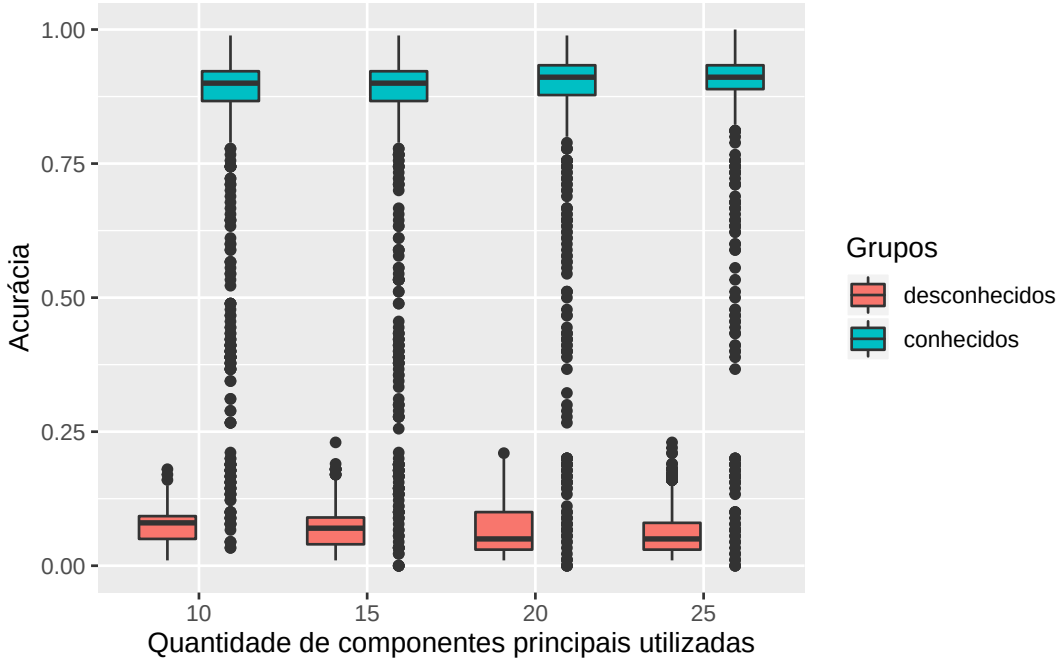


Gráfico 10: Acurácia estimada sob o quantil 0.40

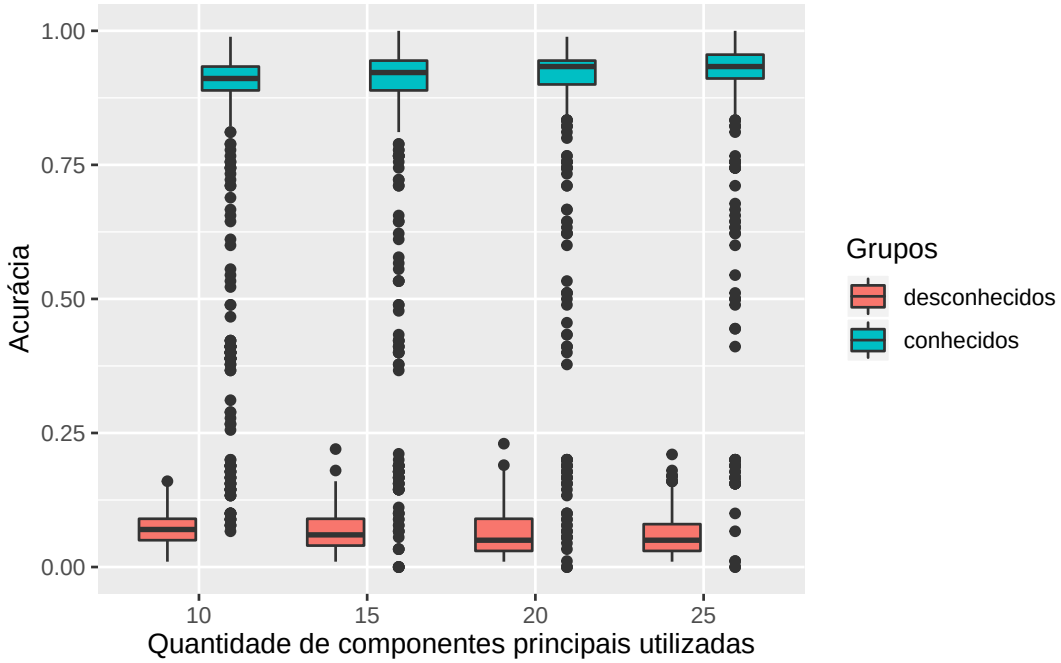


Gráfico 11: Acurácia estimada sob o quantil 0.45

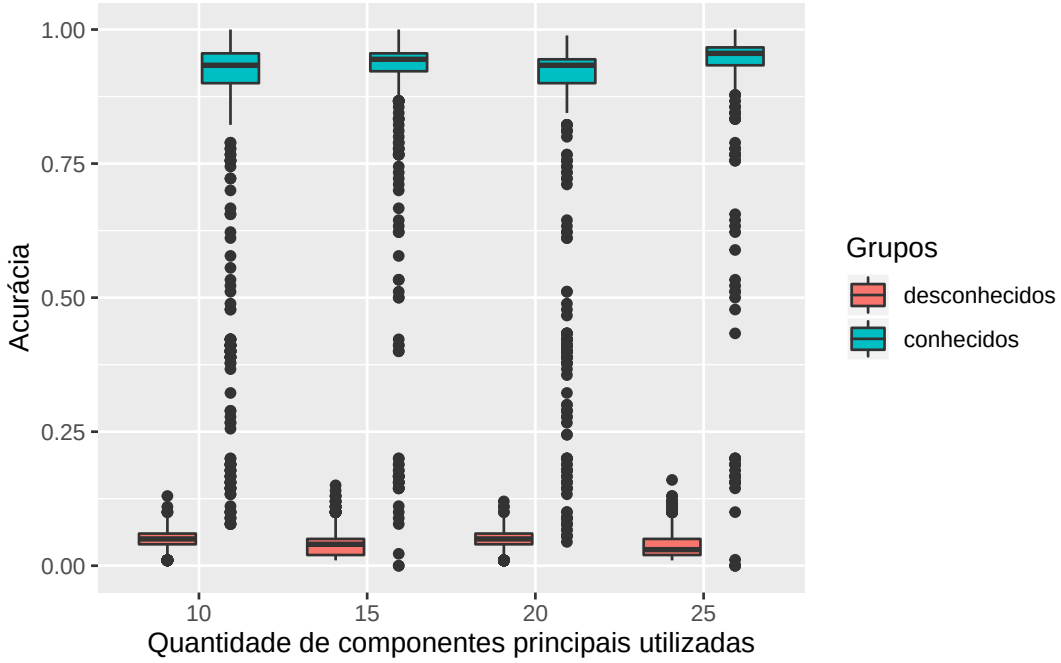


Gráfico 12: Acurácia estimada sob o quantil 0.50

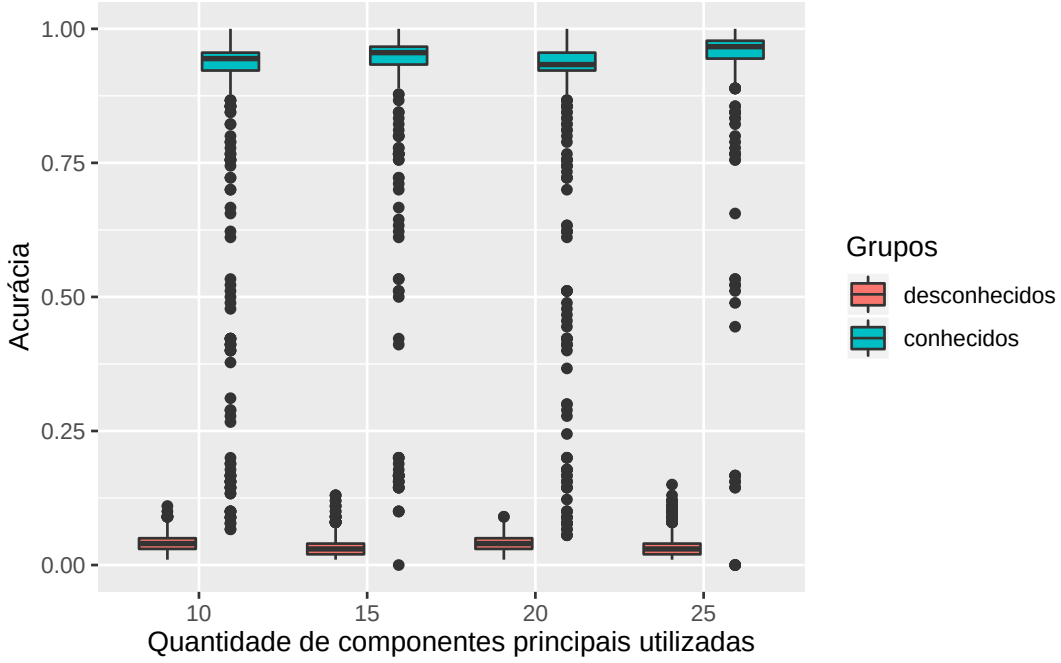


Gráfico 13: Acurácia estimada sob o quantil 0.55

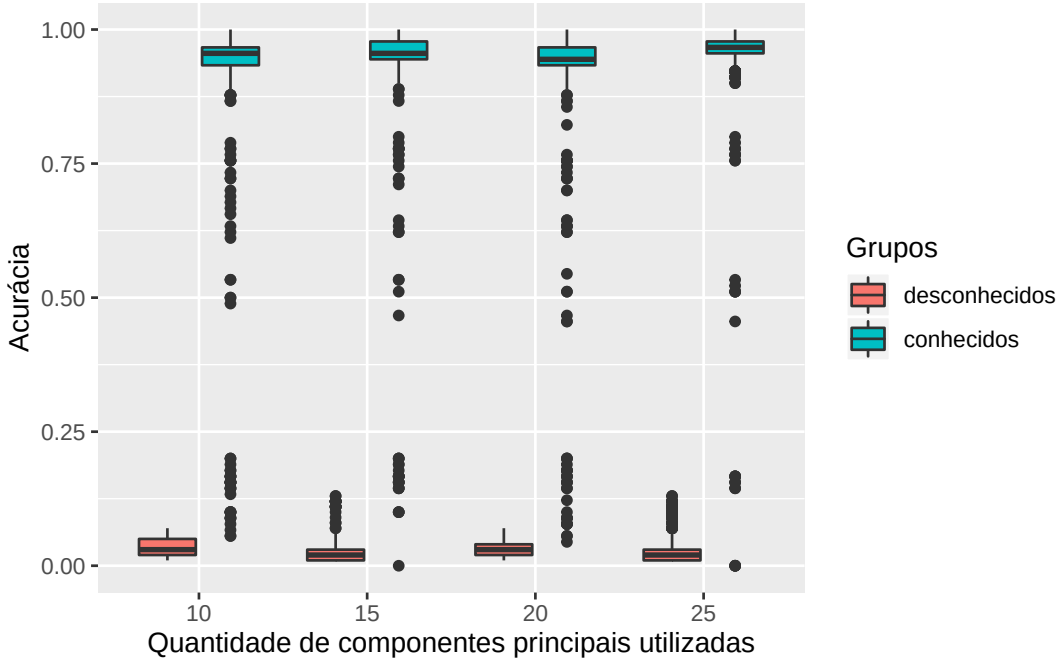


Gráfico 14: Acurácia estimada sob o quantil 0.60

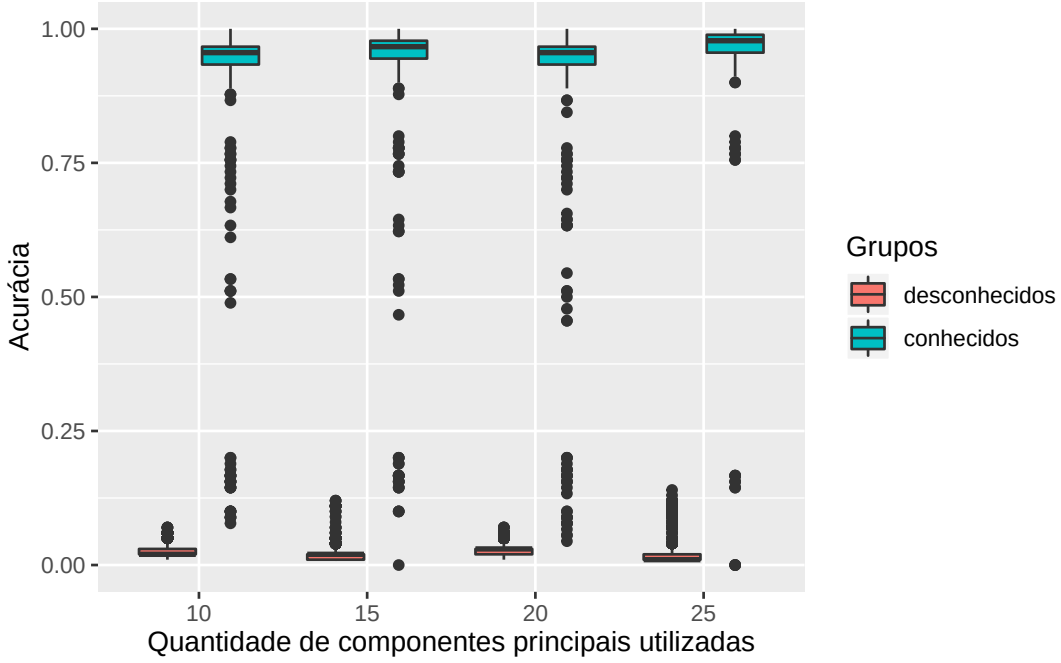


Gráfico 15: Acurácia estimada sob o quantil 0.65

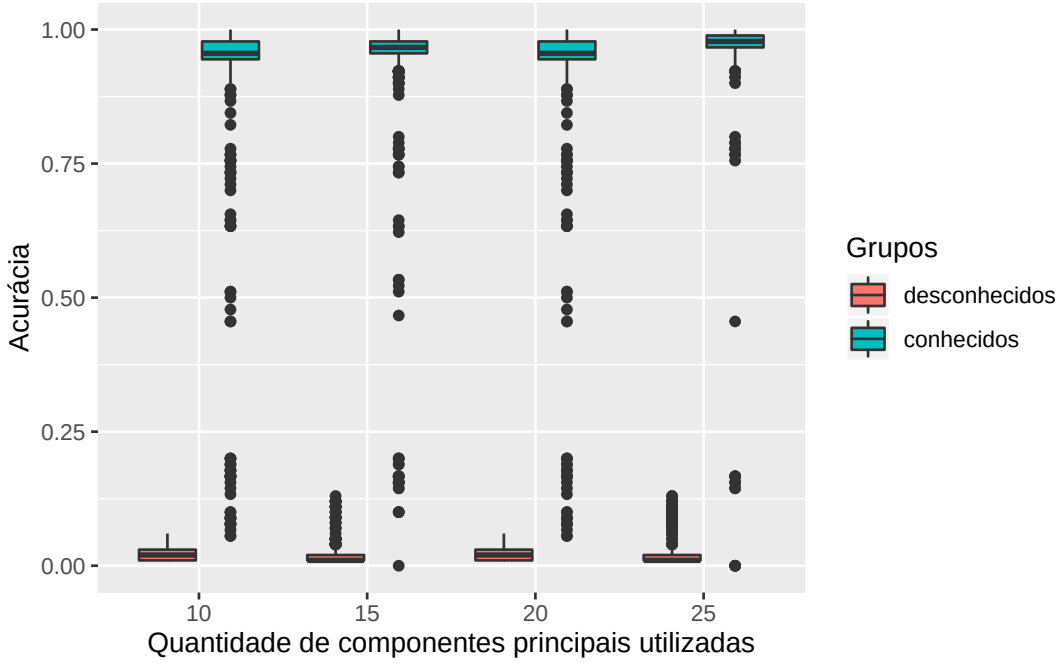


Gráfico 16: Acurácia estimada sob o quantil 0.70

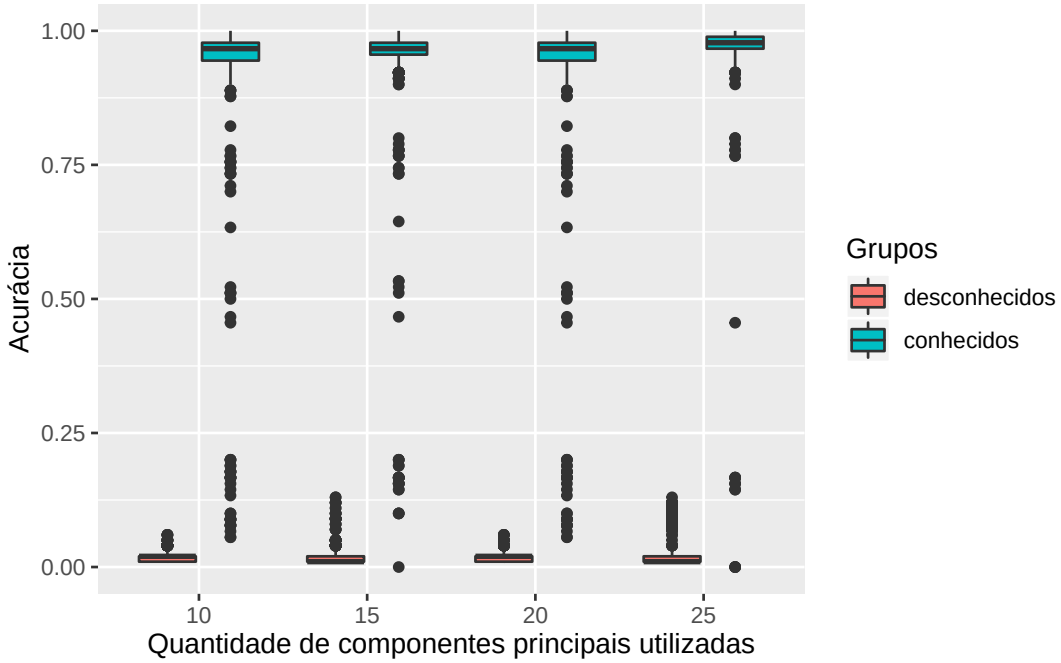


Gráfico 17: Acurácia estimada sob o quantil 0.75

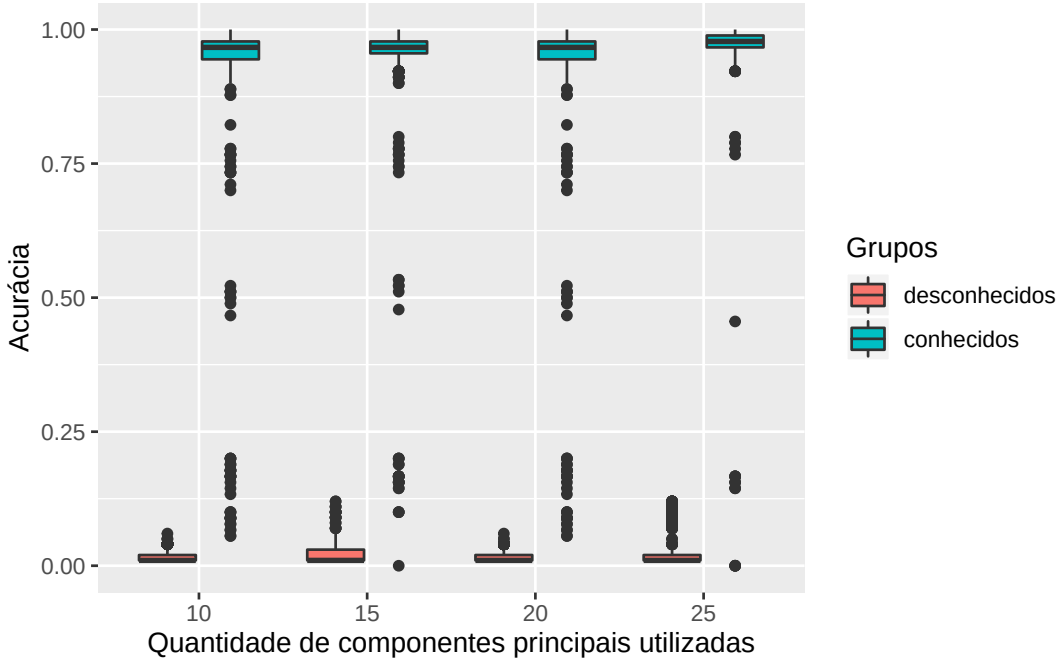


Gráfico 18: Acurácia estimada sob o quantil 0.80

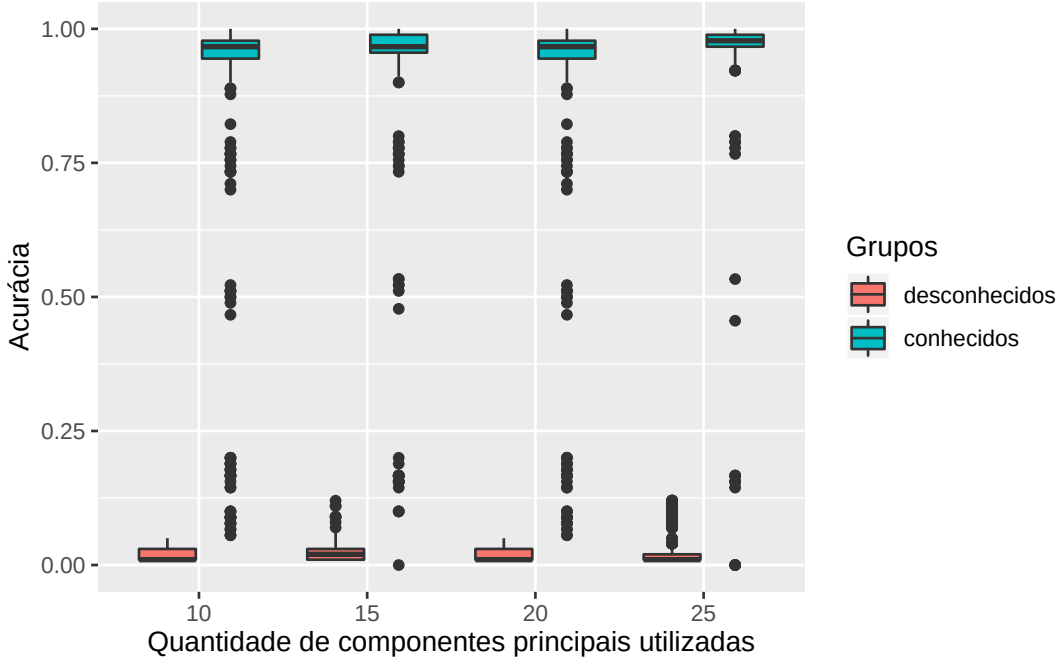


Gráfico 19: Acurácia estimada sob o quantil 0.85

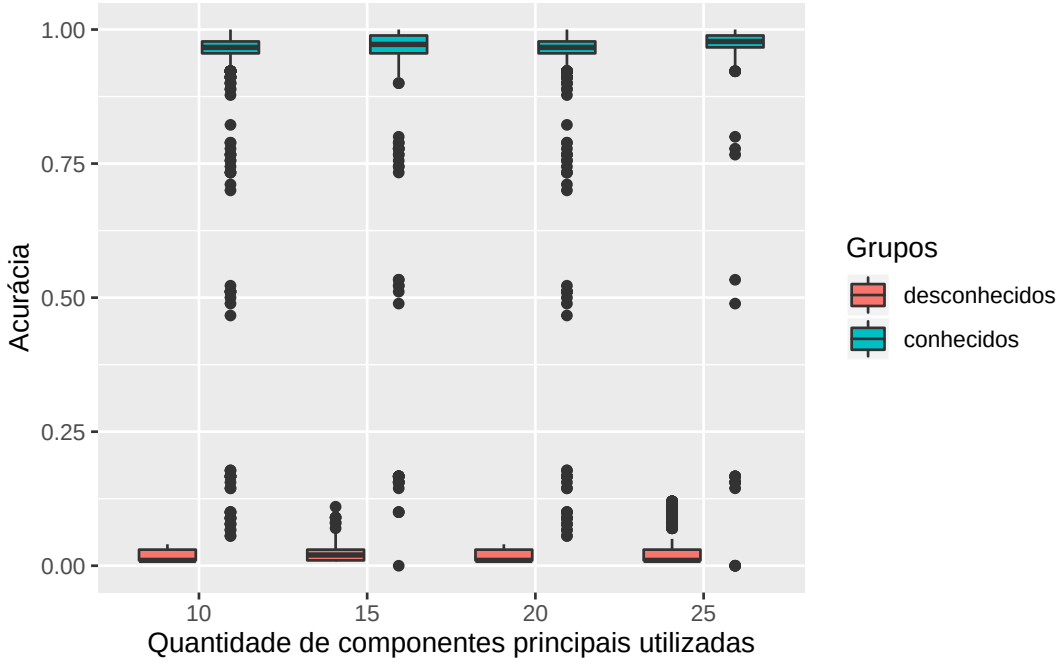


Gráfico 20: Acurácia estimada sob o quantil 0.90

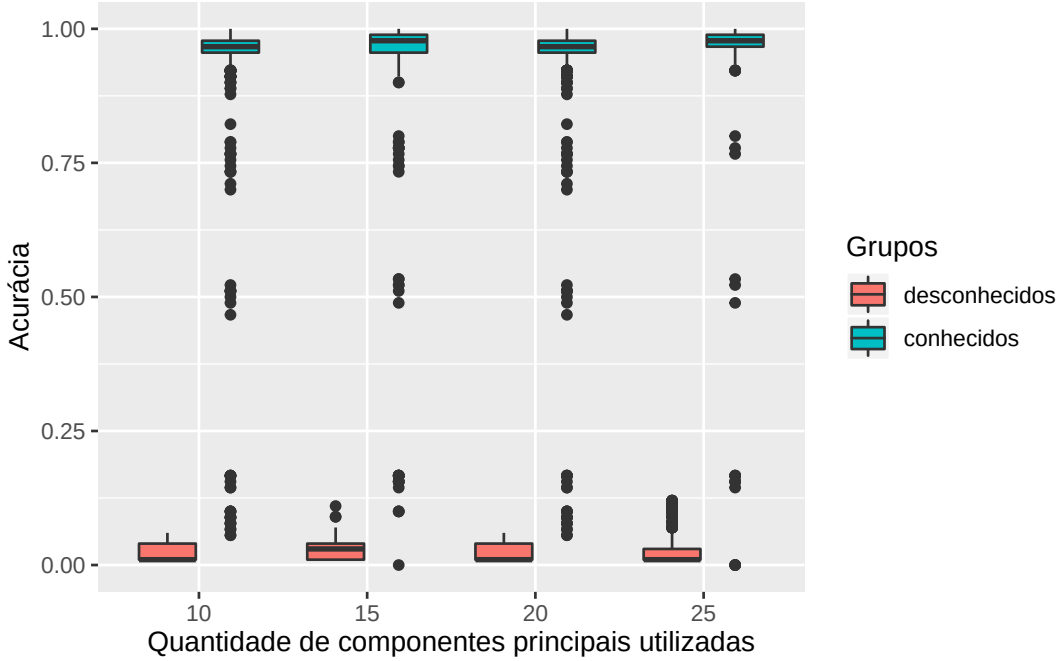
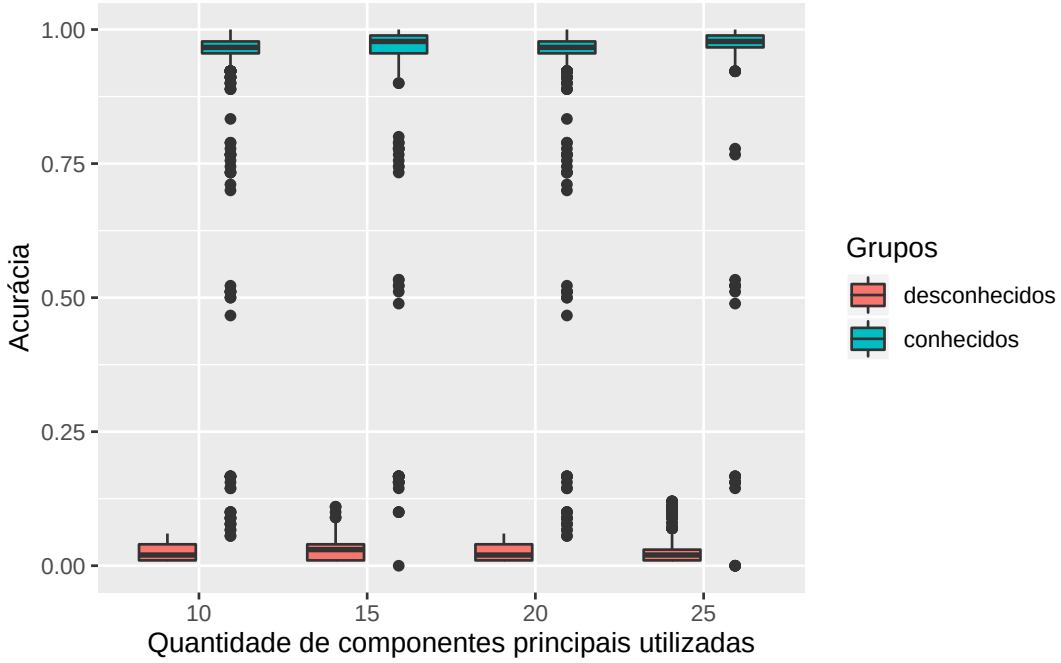


Gráfico 21: Acurácia estimada sob o quantil 0.95



Considerando esse caso de teste, no qual o classificador deve estipular limites de aceitação para cada classe, os resultados não foram satisfatórios, mostrando que para esse cenário é interessante a utilização de técnicas mais sofisticadas. O emprego de diferentes quantidades de componentes principais não implicou em variações significativas na taxas de acertos. Ao utilizar limites mais rígidos, o classificador consegue rejeitar mais imagens desconhecidas, porém não classifica corretamente aquelas imagens que pertençam a algum grupo. Caso os limites se tornem mais abrangentes, o algoritmo não consegue identificar as imagens desconhecidas.

7 Conclusão

A utilização de reconhecimento biométrico é uma tecnologia que traz mais segurança ao processo de identificar pessoas. Dentre os vários tipos de características físicas utilizadas para essa finalidade, optou-se por trabalhar com reconhecimento facial devido a facilidade de emprego da técnica e disponibilidade de bases de dados para realização do experimento.

Outros trabalhos já foram realizados e serviram de base para o desenvolvimento desse. Além de realizar a classificação de imagens de grupos utilizados para treinamento do classificador, foi proposta uma versão que fosse capaz de identificar imagens que não pertençam a nenhuma classe a partir da distribuição empírica das distâncias entre as imagens de cada grupo treinado.

Em razão do espaço vetorial de alta dimensão no qual imagens são representadas, foi aplicada a técnica de análise de componentes principais para representar essas imagens em um subespaço vetorial de dimensão significativamente inferior ao original.

Em relação aos resultados obtidos, para o primeiro caso, no qual o algoritmo classifica apenas imagens de grupos utilizados no treinamento, foi obtida uma taxa de acertos alta, na faixa de 95%. Para o caso em que o classificador deveria alocar imagens desconhecidas em um grupo a parte, o algoritmo não se mostrou adequado para essa finalidade.

É importante ressaltar que o método de classificação aplicado aqui apresenta uma ideia bastante intuitiva inclusive e que apesar de simples, apresenta ótimos resultados se empregado em casos de teste bem controlados.

Por fim, podemos considerar para próximos estudos a aplicação de técnicas mais sofisticadas de classificação, como Support Vector Machines e Redes Neurais.

A Códigos utilizados

```r

classificador <- function(dados,treino,cp){

  # separar base de treinamento
  dt.treino <- dados[treino,]
  dt.teste <- dados[-treino,]

  # indice da ultima coluna - serve para retira-la para rodar a PCA
  # a ultima coluna possui as classes das imagens
  last <- ncol(dados)

  # Aplicacao da PCA nos dados de treinamento
  componentes <- prcomp(dt.treino[,-last], center = T, scale. = T)
  resumo <- summary(componentes)

  # separacao dos autovetores
  auto.vetores <- resumo$rotation[,1:cp]

  # calculo dos escores
  escores.treino <- as.matrix(dt.treino[,-last])%*%auto.vetores
  escores.teste <- as.matrix(dt.teste[,-last])%*%auto.vetores

  # Matriz para armazenamento das distancias
  distancias <- matrix(0,nrow = nrow(escores.teste),
                       ncol = nrow(escores.treino))

  # Calculo das distancias
  for(i in 1:nrow(distancias)){
    for(j in 1:ncol(distancias)){
      distancias[i,j] <- sqrt(sum((escores.teste[i,]-
                                   escores.treino[j,])^2))
    }
  }

  # Vetor com as menores distancias em cada linha
  resultado <- apply(distancias,1,which.min)

  # Vetor para armazenar as classificacoes corretas e erradas
  acertos <- vector()

  # Classificacao das imagens
  for(i in 1:length(resultado)){
    if(dt.teste[i,last]==dt.treino[resultado[i],last]) acertos[i] = 1
    else acertos[i] = 0
  }

  # Acuracia
  acuracia <- sum(acertos)/length(resultado)

  return(acuracia)
}
```

Listing 1: Classificador

```
1
2  classificador2 <- function(indices){
3
4    # Separacao da base de treinamento e de teste
5    treino <- dados1[indices,]
6    teste <- dados1[-indices,]
7    teste2 <- dados2
8
9    # Numero da   ltima   coluna
10   last <- ncol(treino)
11
12   # PCA
13   componentes <- prcomp(treino[,-last], center = T, scale. = F)
14   resumo <- summary(componentes)
15
16   # separacao dos autovetores
17   auto.vetores <- resumo$rotation[,1:15]
18
19   # calculo dos escores
20   escores.treino <- as.matrix(treino[,-last])%*%auto.vetores
21   escores.teste1 <- as.matrix(teste[,-last])%*%auto.vetores
22   escores.teste2 <- as.matrix(teste2[,-last])%*%auto.vetores
23
24   # definicao dos limites por classe
25   aux <- 30*7
26   ref <- matrix(1:aux,nrow = 7)
27   limites <- data.frame(classe = c(1:30),limite = rep(0,30))
28
29   for(i in 1:ncol(ref)){
30     limites$limite[i] <- quantile(dist(escores.treino[ref[,i],],method
         = "euclidian"),probs = 0.95)
31   }
32
33   # Matriz para armazenamento das distancias
34   distancias1 <- matrix(0,nrow = nrow(escores.teste1), ncol = nrow(
       escores.treino))
35
36   for(i in 1:nrow(distancias1)){
37     for(j in 1:ncol(distancias1)){
38       distancias1[i,j] <- sqrt(sum((escores.teste1[i,]-escores.treino[j
           ,])^2))
39     }
40   }
41
42   # Matriz para armazenamento das distancias
43   distancias2 <- matrix(0,nrow = nrow(escores.teste2), ncol = nrow(
       escores.treino))
44
45   for(i in 1:nrow(distancias2)){
46     for(j in 1:ncol(distancias2)){
47       distancias2[i,j] <- sqrt(sum((escores.teste2[i,]-escores.treino[j
           ,])^2))
48     }
49   }
50
51   resultado1 <- data.frame(obs = c(1:nrow(teste)),
52                            real = teste[,last],
```

```
                              ajustado = rep(31,nrow(teste)))

  for(i in 1:nrow(distancias1)){
    # data.frame para ordenar dist ncias
    dist1 <- data.frame("distancia" = distancias1[i,],"classe" = rep
        (1:30,each = 7))
    dist2 <- dist1[order(dist1$distancia),]

    # teste das dist ncias
    k <- 1
    ok <- 0
    while(ok == 0 & k <= ncol(distancias1)){
      if(dist2[k,1] <= limites$limite[dist2[k,2]]){
        resultado1[i,3] <- dist2[k,2]
        ok <- 1
      }
      k <- k + 1
    }
  }

  resultado2 <- data.frame(obs = c(1:nrow(teste2)),
                           real = teste2[,last],
                           ajustado = rep(31,nrow(teste2)))

  for(i in 1:nrow(distancias2)){
    # data.frame para ordenar dist ncias
    dist1 <- data.frame("distancia" = distancias2[i,],"classe" = rep
        (1:30,each = 7))
    dist2 <- dist1[order(dist1$distancia),]

    # teste das dist ncias
    k <- 1
    ok <- 0
    while(ok == 0 & k <= ncol(distancias1)){
      if(dist2[k,1] <= limites$limite[dist2[k,2]]){
        resultado2[i,3] <- dist2[k,2]
        ok <- 1
      }
      k <- k + 1
    }
  }

  tabela1 <- table(resultado1$real,resultado1$ajustado)
  acuracia1 <- sum(diag(table(resultado1$real,resultado1$ajustado)))/
      nrow(teste)

  tabela2 <- table(resultado2$real,resultado2$ajustado)
  acuracia2 <- sum(diag(table(resultado2$real,resultado2$ajustado)))/
      nrow(teste2)

  return(list(acuracia1 = acuracia1,
              resultado1 = resultado1,
              acuracia2 = acuracia2,
              resultado2 = resultado2))

}
```

Listing 2: Classificador 2

Referências

ATT LABORATORIES CAMBRIDGE. The Database of Faces. Disponível em: <https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>. Acessado em 17 de março de 2019.

JAMES, Gareth et al. An introduction to statistical learning. New York: springer, 2013.

JOHNSON, Richard Arnold et al. Applied multivariate statistical analysis. Upper Saddle River, NJ: Prentice hall, 2002.

KIM, Kyungnam. Face recognition using principle component analysis. In: International Conference on Computer Vision and Pattern Recognition. 1996. p. 591.

MANLY, Bryan FJ. Multivariate statistical methods a primer. 2016.

Max Kuhn. Contributions from Jed Wing, Steve Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, Zachary Mayer, Brenton Kenkel, the R Core Team, Michael Benesty, Reynald Lescarbeau, Andrew Ziem, Luca Scrucca, Yuan Tang, Can Candan and Tyler Hunt. (2019). caret: Classification and Regression Training. R package version 6.0-84. https://CRAN.R-project.org/package=caret

MINGOTI, Sueli Aparecida. Análise de dados através de métodos de estatística multivariada: uma abordagem aplicada. Editora UFMG, 2005.

R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/

Simon Barthelme (2019). imager: Image Processing Library Based on 'CImg'. R package version 0.41.2. https://CRAN.R-project.org/package=imager

XU, Qing-Song; LIANG, Yi-Zeng. Monte Carlo cross validation. Chemometrics and Intelligent Laboratory Systems, v. 56, n. 1, p. 1-11, 2001.